CBI 12/3 /41. F.6.13

PROGRESS REPORT NUMBER 9

of the

Research and Educational Activities

in

Machine Computation

by the

Cooperating Colleges of New England

July 1961

Computation Center
Massachusetts Institute of Technology
Cambridge 39, Massachusetts

I. RESEARCH ACTIVITIES

1.0 Introduction

The current research investigations being undertaken by members of the Computation Center staff lie in three major areas: numerical analysis, artificial intelligence, and multi-console time-sharing.

1.1 A Time-Sharing System for the IBM 709

F.J. Corbató
M.L. Merwin
R. Daley

A time-sharing system has been designed and is being debugged. This system makes use of the two Flexowriters and the clocks that have been attached, under the direction of H.M. Teager, to the Direct Data Connection of the IBM 709 on Channel D. Outwardly, the computer may have three "users"; one user is the background FMS system; the other two users are those at the Flexowriters. The Flexo users will direct the operation of the computer by typing in commands. This is partially analogous to the control cards in the FMS system. In addition, the Flexo user has the ability to type in a symbolic program at the Flexowriter and perform some editing on it, if necessary.

Ideally, the information being typed in would be stored on a disc in some area assigned to that particular user. The user's area of disc would probably contain his "program" or collection of subprograms, perhaps, both symbolic and binary. As a substitute for the disc area, tapes are being used in the present version of the system. Each Flexo user has a private tape that is used for his collection of subprograms and data. The tape may be made up off-line or be formed completely from information typed in at the Flexowriter and from any translations that were commanded.

The time-sharing system, TSS, is composed of three major parts. First, the TSS Supervisor and Trap Coordinator control the over-all operation, the acceptance of input and output, and scheduling; they occupy the first 5000 locations in core memory. Second, the background FMS system is a reorigined system starting at 5000 and contains the FAP assembler and the MAD compiler; FORTRAN is not present in this system, but the MADTRAN translator will be available in the system. MADTRAN converts FORTRAN source programs to MAD source programs. Hence, externally the user is coding in FORTRAN, but internally MADTRAN turns the program into MAD and the translation takes place on this MAD source program. Third, the programs that perform certain commands given by the Flexo users make up, with the TSS Supervisor and Trap Coordinator, the TSS system tape. For example, special versions of the translators FAP and MAD will be available to the Flexo users on the TSS system tape.

This complex of TSS programs becomes activated for the first time when a key is hit on the Flexowriter. The background FMS system is interrupted long enough to bring in from the TSS system tape the TSS Supervisor and the Trap Coordinator, place Channel D in the read-in status ready to accept further characters, and activate the clocks. When the carriage return key on the Flexowriter is hit, the line of information that has been typed in is examined for a command. If it is a command, then the TSS Supervisor notes the fact and determines whether to stop running the present user and start the command or to continue the present user for another clock cycle. Present policy is for the TSS Supervisor to give priority to the Flexowriter users; the FMS background user will be run only when both Flexowriters are inactive.

Some of the basic commands are Login, Input, File, FAP, Load, Start, PM, and Logout. These commands represent a simple but typical computer run. A user is logged in at a given time, information is typed in and later processed and filed on his tape, the input is assembled in FAP and loaded with other subprograms perhaps, the program is started, post-mortem information is requested and given, and finally the user is logged out and billed for the computer time used.

Most of the programs for the system are semi-independent and therefore can be assigned as individual projects. The status of these projects is as follows:

The TSS Supervisor and Trap Coordinator represent the major programming effort. The Trap Coordinator is responsible for accepting characters from both typewriters and routing them to the appropriate individual buffers, handling write-out of messages on both typewriters which means the merging of messages from the individual buffers into the write-out buffer, and coordination of simultaneous reading and writing of the Flexos. This program is sufficiently debugged so that it can be relied on in testing of the TSS Supervisor.

The Load command is being written and is a specialized version of the BSS loader.

The File command is really an input processor that treats tabs, backspaces, and underlines (delete line up through this point) correctly according to the syntax of one of the input languages. The input lines from the Flexowriter are automatically numbered in increments of 10. In a special edit mode, the Flexo user numbers his own lines. The File command program checks all sequence numbers and forms a final input set with sequential line numbers and ignores all but the last of the lines having the same number. Thus, simple replacement of lines is possible, as well as the insertion of 9 lines. This program is being written and is not debugged.

The Login and Logout commands are operating correctly. These commands do the time accounting for the Flexo users and interact with the background FMS system so that the FMS user is not overcharged for time that must be billed to the Flexo user.

The Listf command is operating. This command reads the titles of every file on the Flexo user's private tape and types the titles on the Flexowriter.

Among the other projects that are in progress but not completed are: 1) incorporation of MAD in both the background FMS System Tape and the TSS System Tape, 2) placing of FAP on the TSS System Tape, and 3) revision of (F2PM) to be the basis of a Post-Mortem command for the Flexo user.

1.2 Real-Time, Time-Shared Computer Project

H.M. Teager
B.W. Munday
M. Kudlick
M. Kovarik

During the past quarter, work has been focused on the following areas, each of which will be described in greater detail in this report:

a) Continuation of systems study of on-line languages.

b) Continuation of "program analysis" type of programming systems.

c) Continuation of programming for memory and time allocation in time-shared usage.

d) Construction and checkout of first generation input-output devices.

e) Continuation of studies and experimentation of second generation user consoles.

f) Agreement with IBM on design changes to 7090 system for time sharing.

g) Development of programming systems to second generation devices.

Progress has been steady and in accordance with the schedule of the project.

1.2.1 On-Line Languages

With the concurrent development of low-cost graphical input and output devices to be used in real time, major improvements in programming languages are now becoming feasible and are being explored.

As has been stated in previous reports, graphical devices open up the possibility for multidimensional pictorial and symbolic languages, whereas keyboard devices restrict the user to essentially one-dimensional languages.

Three classes of graphical languages are presently being explored.

a) An algorithmic language in the form of flow charts for describing processes.

b) A geometric language in the form of dimensional diagrams and equations for describing physical systems governed by field equations.

c) A structural language for describing symbolic networks.

At the present stage, few conclusions have been drawn. However, it appears that algorithmic languages, such as Algol and Cobol, are far more readily understood by the programmer when the flow aspects of the algorithm are shown by arrows and boxes, and the formula aspects are shown by ordinary mathematical formulas using subscripts, superscripts, and other two-dimensional notations rather than parentheses.

A Master's Thesis has been done in the area of geometric languages for describing field problems. Procedures for converting the language into efficient computation algorithms were also explored.

Several types of networks (such as encountered in electrical systems or neural models) have been studied with a view of development of a structural language.

Attempts in all three areas are as yet exploratory and not well defined.

1.2.2 Program Analysis

It has been clear for some time that it would be highly desirable to be able to modify the structure of programs (such as compilers) by machine in order to meet economically objectives that were not considered by the designers of the programs.

To this end, a programming system for analysis and reconstitution of programs is being developed. The first objective of the system is to ive the human programmer such reference information as he would need in rder to make changes and reorganizations in existing programs. This program has been tested and is being used.

The reorganization that is necessary for effective on-line time-sharing of presently existing compilers must provide for:

a) Substitution of real-time input-output devices for standard card readers and printers, with appropriate language and program conventions.

b) Modification of all diagnostic messages and routines to allow on-line correction in the appropriate source language without loss of computer time.

c) Modification of compilers to limit the amount of contiguous program necessary in memory at any stage.

d) Modification of compilers so as to produce object programs which limit the amount of memory needed (at any time) for object programs.

e) c and d to be accomplished by using disc file "backup memory" to minimize loss of machine efficiency in replacing program segments.

f) Modification of object programs to allow human interaction.

At present, the principal compilation system undergoing study is the MAD (Michigan Algorithmic Decoder) system. The modifications to this system are now well understood and are being made.

It is presently hoped to have time-shared versions of this system working by the end of 1961.

1.2.3 Allocation Programming Systems

A programming system for scheduling time-shared users, determining internal memory and disc storage allocation and access, and retrieval for information is being programmed and simulated.

The system is being designed for the modified IBM 7090 expected at the Center early in 1962, but vital portions of that system (such as the disc files) will not be installed until later in 1962 and will be simulated with tape.

Many detailed decisions with respect to this system strongly hinge upon the languages, devices, and programming systems that will be available for time sharing; thus, for the present, the system is being kept as versatile as possible. All of the major problems involved in memory and time allocation (subject to a minor constraint) have now been satisfactorily solved. The constraint is a known upper limit on contiguous memory for each program segment.

1.2.4 Construction and Checkout of First Generation Input-Output Devices

A buffer unit for two Flexowriters, photoelectric tape reader, and other devices has been built and checked out. It is now undergoing final testing with the present 709 computer.

The typewriters are capable of remote cable connection, and usage tests will be made in this mode as soon as programming systems are ready.

The buffer unit incorporates other needed hardware features such as interval timers for program scheduling.

1.2.5 Second Generation Real-Time Consoles

During the past quarter, work has been concentrated on design studies, experimentation for graphical input devices, and feasibility studies in graphical output devices.

Several types of graphical input devices have been studied, and a prototype is being constructed.

The display devices of many different manufacturers have been explored from a viewpoint of cost, display storage time, data rate, data capacity, and resolution. Although there are now well over twenty manufacturers in this field, none of the equipment is yet suitable from a systems viewpoint. Costs are high, and capabilities are in general too low.

There is a third aspect of the graphical input-output question that is being explored, namely, the generation of a permanent record. It is possible that graphical output, if in the form of an electronic digital plotter, could provide both display and permanent records.

1.2.6 Design Changes to 7090

Agreement has been reached with IBM with respect to several design modifications for time-sharing the 7090 computer. These modifications include program relocation and memory protection features. The modifications are to be included in the 7090 to be delivered in early 1962.

1.2.7 Programming Systems for Second Generation Input-Output Devices

Programming systems are being developed for recognition of handwritten symbols and marking developed by the graphical input device.

It is as yet too early for similar systems for the graphical outputs.

1.3 Artificial Intelligence

J. McCarthy
M.L. Minsky
P.W. Abrahams
B.H. Bloom
D.G. Bobrow
D. Edwards
T. Hart
L. Hodes
M. Levin
D. Luckham
D. Park
B. Raphael

1.3.1 Algebraic Simplification and Electrical Network Solution Using LISP

An algebraic simplification program has been written in LISP and used in a program which finds input and transfer impedances of an electrical network with linear, symbolic, or numeric valued elements. The algebraic simplification program makes all possible simple reductions on an algebraic expression such as actually adding up numeric terms in a sum, finding a common denominator for a sum of fractions, collecting common subexpressions from sums of products, and symbolically multiplying out a product of sums.

The network solving program uses heuristic searches for transformations on the network which reduce the number of nodes, in order to simplify the symbolic computations which must be made. The network solver incorporates routines for setting up and solving a floating admittance matrix for the network system functions desired.

Experiments with the network solver using the LISP interpreter indicate that its simplification and the heuristic components are quite efficient. However, the floating admittance matrix reduction technique requires too much unnecessary symbolic algebraic manipulation to be practical for use by electrical engineers.

T. Hart

1.4 Chebyshev Approximation

P. Fox
A.A. Goldstein

An algorithm for certain problems of nonlinear Chebyshev approximation is being applied to the problem of numerical rational approximation.

M1401. A DYNAMIC MODEL FOR EVALUATING THE WORTH OF A DEVELOPMENT PROJECT

Simulation of factors affecting return on a development project - major emphasis on development, marketing, production and competition.

Joseph F. Rocky
School of Industrial Management

1402. COMPUTER SOLUTIONS OF PROBLEMS IN APPLIED NAVAL ARCHITECTURE

The proposed work will consist of two major parts:
Preparing a program for extrapolating resistance from a model to a geo-
etrically similar ship, using input data having any engineering form in current
use, at the discretion of the eventual user. It will utilize the usual Froude
assumption. Various friction formulations will be used and compared. The pro-
ram will be able to reduce and analyse the data from resistance experiments at
the MIT towing tank.
Preparing a program to utilize the Evans method for the rational design
transversely framed midship sections. The program will have to choose the
applicable criterion of failure for each structural member of an arbitrary ship,
and design against such failure. An attempt will be made to optimize the moment
inertia of the ship with respect to the weight of the structure.

Dan Khoushy
Naval Architecture

M1403. A PROGRAM TO PLAY CONTRACT BRIDGE -- BIDDING ONLY

One of the most popular of all card games is contract bridge. As a
game having many of the challenges of chess while being somewhat more systematiz-
able, it is an ideal target for a computer program. To make the project more
manageable, the program will be limited to the bidding (auction), one of the two
subdivisions of the game. The actual playing of the hands may be programmed.
The main problem will be to develop appropriate procedures so that it
is impossible to state at this time what procedures will be used. Various pro-
cedures will be tested by comparing machine bids with expert opinion; mostly
in the form of games already played by experts.

Gay L. Carley
Electrical Engineering

M1404. ELE

We
which are un
canonical fo
remapped as
by an approp
counts. The

M1405. UTI
LUT

It
The program
symmetric de
supported an
analytical s
clamped at t

M1406. PRO

Th
the program
to place ca
because the
priority to
It is hoped
can eventua

N1440. PHYS. 113 - COURSE IN PROGRAMMING FOR HIGH-SPEED DIGITAL COMPUTERS

A course in general programming for about forty students. FAP and FORTRAN are used.

R. H. Bretton
Yale University

M1441. A BRIDGE-PLAYING PROGRAM

This program will be written to satisfy the term paper requirements of 18.16, Automata and Artificial Intelligence. The complexity of bridge has led many masters of the game to develop heuristics for bidding and playing. We will use these heuristics and attempt to extend them to the point where they are usable in a machine program which will play good bridge.

Allan L. Scherr and Jospeh P. Wright
Mathematics and Electrical Engineering

M1442. RESPONSE OF SHORT REINFORCED BEAM COLUMN TO ECCENTRIC DYNAMIC LOAD

Investigation of various failure criteria such as maximum tolerable strain, and maximum tolerable deflection will serve as a basis for studying the three modes of failure; namely, tension, compression, and balance design.

Tahir Ahmad Karamat
Civil Engineering

M1443. EXPANSION OF GAS INTO A VACUUM

A set of ordinary differential equations, arising from a similarity solution for the collapse of a gas into a spherical cavity, will be integrated using the Runge-Kutta method.

D. S. Butler
Department of Mathematics